

Offline password cracking

We might find passwords or other credentials in databases. These are often hashed, so we need to first identify which hash it is and then try to crack it. The first step is to identify the hash-algorithm that was used to hash the password.

Identify hash

There are generally speaking three pieces of data we can use to identify a hash. - The length of the hash - The character set - Any special characters

In order to identify a hash we can either use specialized tools that analyze the hash and then return a guess on which algorithm it is. An easier way is of course to just look in the documentation of the software where you found the hashes. It usually says in the documentation or the source code which type of hash is being used.

In kali we can use `hash-identifier` or `hashid` :

```
hash-identifier
hashid
```

Or try these online services:

<http://www.onlinehashcrack.com/hash-identification.php>

https://md5hashing.net/hash_type_checker

Windows domain - Checking password complexity

Find out the password complexity in windows.

```
net accounts /domain
net accounts
```

Cracking the hash

Okay so now we know what hash it is, let's get cracking.

If you want to try out the functionality of hashcat or john the ripper you can find example hashes here: <http://openwall.info/wiki/john/sample-hashes>.

Hashcat

Look for the specific type of hash you want to crack in the list produced by the following command:

```
hashcat --help
```

My hash was a Apache md5, so I will use the corresponding code for it, `1600`

```
-a 0 - straight
```

```
-o found.txt - where the cracked hash outputs
```

``admin.hash"` - the hash you want to crack.

```
/usr/share/hashcat/rules/rockyou-30000.rule - the wordlist we use
```

```
hashcat -m 11 -a 0 -o found.txt admin.hash /usr/share/hashcat/rules/rockyou-30000.rule
```

John the ripper

So this is how you usually crack passwords with john

```
john --wordlist=wordlist.txt dump.txt
```

If you do not find the password you can add the john-rules. Which add numbers and such things to each password.

```
john --rules --wordlist=wordlist.txt dump.txt
```

Linux shadow password

First you need to combine the passwd file with the shadow file using the unshadow-program.


```
unshadow passwd-file.txt shadow-file.txt > unshadowed.txt
john --rules --wordlist=wordlist.txt unshadowed.txt
```

Rainbow tables

So basically a rainbow table is a precalculated list of passwords. So instead of having to hash the word you want to try you create a list of hashes. So you do not have to hash them before comparing. This might take a long time to do, hashing a whole wordlist, but when you do the comparison between the password and the test-word it will go a lot faster.

SPN - Kerberoasting

```
./hashcat-5.0.0/hashcat64.bin -a 0 -m 13100 -o result.txt -r ./hashcat-5.0.0/rules/dive.rule ./hashar.txt /usr/share/wordlists/rocky
```



Using Online Tools

findmyhash

You can use findmyhash

Here is an example of how to use it:

```
findmyhash LM -h 6c3d4c343f999422aad3b435b51404ee:bcd477bfdb45435a34c6a38403ca4364
```

Cracking

Crackstation <https://crackstation.net/>

Hashkiller <https://hashkiller.co.uk/>

Google hashes Search pastebin.

Windows

If you find a local file inclusion vulnerability you might be able to retrieve two fundamental files from it. the `system` registry and the `SAM` registry. There two files/registries are all we need to get the machines hashes. These files can be found in several different locations in windows. Here they are:

```
Systemroot can be windows
%SYSTEMROOT%\repair\SAM
windows\repair\SAM
%SYSTEMROOT%\System32\config\RegBack\SAM

System file can be found here
SYSTEMROOT%\repair\system
%SYSTEMROOT%\System32\config\RegBack\system
```

So if the manage to get your hands on both of these files you can extract the password hashed like this:

```
pwdump system sam
```